SYSTEMS AND METHODS FOR MONITORING RESOURCE UTILIZATION AND APPLICATION PERFORMANCE

Background of the Invention

[0001] A significant challenge in the modern datacenter is to ensure that enough resources, such as computer servers, are available to each application or application component in the data center when there is a sudden peak in demand for the application. The problem becomes compounded for aggregate application, which are deployed on a multi-tier architecture, requiring one or more servers for each tier. For example, an aggregate application may include one or more Web servers for clients of the application to connect to, a set of applications servers that implement the business logic of the application, and one or more database servers to manipulate the data required by the application. Enough resources must be provided at each tier to meet the performance goals of the aggregate application.

[0002] Multiple application components may also be consolidated on a shared resource. This makes it possible to increase utilization of the shared resource. However, application components may be allocated a larger percentage of the shared resource than needed in order to meet peak demand requirements.

Summary of the Invention

[0003] Systems and methods are disclosed for monitoring application resource utilization and application performance. In one embodiment, a system comprises resource data, application data, policy data, and a monitoring agent. The resource data includes information on a plurality of resources and the resources include a plurality of computers. Application data includes one or more application profiles. Each of the application profiles has a performance profile and a resource profile. The resource profile includes resource utilization information. The policy data includes one or more application performance policies and one or more resource utilization policies. Finally, the monitoring agent monitors application data for compliance with the policy data.

[0004] In another embodiment, a method for dynamically monitoring and managing application performance is disclosed. The method includes monitoring application data for compliance with one or more performance policies. The application data includes one or more application profiles, each of which has a performance profile and a resource profile. The resource profile includes resource utilization information associated with an application. The applications each execute in a container associated with a domain. The domains each include one or more resources and the resources include a plurality of computers. In response to a policy violation, the policy is automatically enforced by reallocating how resources are mapped to containers.

Brief Description of the Drawings

[0005] Illustrative embodiments of the invention are illustrated in the drawings in which:

[0006] FIG. 1 illustrates one embodiment of a system for monitoring application performance;

[0007] FIG. 2 illustrates an exemplary grouping of resources into domains that may be monitored by the controller of FIG. 1;

[0008] FIG. 3 is a flow diagram illustrating an exemplary method for enforcing a policy that may be used by the controller of FIG. 1;

[0009] FIG. 4A illustrates an exemplary partitioning of a server into multiple hardware partitions, one of which is container for an application that may be monitored by the controller of FIG. 1;

[0010] FIG. 4B illustrates an exemplary further partitioning of one of the hardware partitions of FIG. 4A into multiple processor sets, one of which is a container for an application;

[0011] FIG. 4C illustrates an exemplary further partitioning of one of the network partitions of FIG. 4A into multiple software-based partitions, one of which is a container for an application;

[0012] FIG. 4D illustrates an exemplary further partitioning of one of the software-based partitions of FIG. 4C into multiple sub-cpu resource partitions, two of which are containers for an application;

[0013] FIG. 5 is a flow diagram illustrating an exemplary method for enforcing a policy that may be used by the controller of FIG. 1 to enforce a policy on the server of FIGS. 4A-D;

[0014] FIG. 6 is a flow diagram illustrating an exemplary method for enforcing a policy that may be used by the controller of FIG. 1 to enforce a policy in a cluster domain; and

[0015] FIG. 7 illustrates one embodiment of a domain implemented in a cluster environment that may be monitored by the controller of FIG. 1.

Detailed Description

[0016] An exemplary system for monitoring application performance across multiple resources is illustrated in FIG. 1. A controller 100 includes resource data 104. Resource data 104 contains information on a plurality of resources. Resources may include a plurality of computers, such as servers, or blades in a rack and blade architecture. Resources may also include other types of compute resources, such as partitions within a server, and other devices, such as load balancers, firewalls, and network switches. By way of example, resource data 104 may include information on the capacity of the resource, the network address of the resource, and how to instantiate (initialize, boot, and configure) the resource.

[0017] Controller 100 also includes application data 110. Application data includes one or more application profiles 112, 114, 116. An application may be a single application, a replicated application, or an aggregate application. A replicated application may be a set of replicated instances of a single application, which together perform a single function. By way of example, a replicated application may be a Web server farm or a replicated database application, such as Oracle®'s Real Application Clusters(RAC). An aggregate application may be a combination of multiple single and/or replicated applications across multiple tiers.

[0018] In one embodiment, application profiles 112-116 may include one or more of a resource profile, and a performance profile. A resource profile may include resource demand information on the amount of resources an application requires and resource utilization information. The resource utilization information may include resource allocation information on the amount of resources an application is currently assigned, and/or resource consumption information on the amount of resources an application is using or has used over a period of time. By way of example, information on CPU, memory, I/O, network percentages or absolute consumption for an application may be tracked over a period of time and stored in a resource profile. A performance profile may include information on application performance at the application or user level, such as response time. Finally, a demand profile may measure the internal activity of an application. It should be appreciated that application profiles 112-116 may also include additional information, such as a relative priority of an application and its components, details on how performance is to be monitored, or instructions on how to install and configure the application.

[0019] In one embodiment, applications may be associated with a container. A container may be a logical computer where an application or application component resides. A container may have its own copy of an operating system, or it might be implemented within an operating system. By way of example, a container may be an unpartitioned server running a single application, a hardware partition, a software-based partition, a processor set, a sub-CPU resource partition (partitions of a single CPU resource), multiple nodes of a cluster, or other set or unit of computer resources.

[0020] Controller 100 may receive performance information for an application profile 112-116 from a plurality of client agents 120, 122, 124. Each client agent may run on an operating system instance on a resource and may monitor the performance of applications running on the operating system instance. It should be appreciated that in alternate embodiments, the performance information for an application profile may be obtained with an alternate method.

[0021] Policy data 108 is also accessible to controller 100. Policy data 108 may include one or more performance policies associated with an application or application component. By way of example, an application policy may be that an average response time per transition for the application component is 2 seconds 95% of the time. Policy data may also include one or more resource utilization policies associated with a resource, an application, or a container. For example, a utilization policy may be that the maximum utilization allowed for a container or a resource is 80%. Other performance and resource utilization policies are also contemplated. Additionally, in some embodiments, one or more of the policies may be assigned a relative priority.

[0022] Controller 100 additionally includes monitoring agent 102 to monitor the application data for compliance with the policy data. In one embodiment, the monitoring agent may provide advisory information about potential actions that can be taken to maintain or restore compliance with application performance or utilization policies. As will be described in further detail below, in other embodiments, the monitoring agent may adjust resources (e.g., allocate, reallocate, or deallocate them) to enforce policies.

[0023] Controller 100 additionally includes domain definition data 106. The domain definition data includes information on one or more domains. Each domain contains a grouping of resources, such as one or more computers or containers, which provide a shared pool to be shared by one or more applications or application components. By way of example, a domain may consist of hyper-text transfer protocol (HTTP) servers, all of which may share the job of providing web access to several applications. The domain definition data may also include resource utilization information for one or more of the domains. In one embodiment, the monitoring agent may monitor the domain resource utilization information and provide advisory information about potential actions that can be taken to maintain or restore resource utilization to comply with domain policies. In other embodiments, the monitoring agent may dynamically adjust resources to enforce domain policies.

[0024] Although FIG. 1 depicts a controller 100 including the various components described above, it should be appreciated that alternate embodiments of these components may be combined or may reside at different physical locations. For example, resource data may reside in a database accessible to controller 100 and application data may reside in a different database. Alternately, domain definition data, resource data, and application data may be combined into one database of information.

[0025] FIG. 2 illustrates an exemplary grouping of resources into domains 200, 210, 220. Domain 200 includes three resources 202, 204, 206. By way of example, resources 202-206 may be servers, nodes in a cluster, blade servers in a rack and blade architecture, or other type of computer resource. Domain 210 includes two resources 212, 214, which may also be servers, nodes in a cluster, blade servers, or other type of computer resource. Domain 220 contains a single resource 222 which, by way of example, may be a nested resource, such as a partitioned Hewlett Packard Superdome computer. Monitoring agent 102 may expand a resource domain if the domain has a need for additional resources or may contract a domain if a domain has extra unused capacity. By way of example, domains may be expanded by using capacity-on-demand-processors or obtaining a server or blade from a staging area. As will be described in further detail below, resources may also be arbitrated (e.g., determining how resources are allocated to containers) across a domain. Expansion and arbitration information may be included in policy data 108, domain definition data 106, or another location.

[0026] In some embodiments, controller 100 may perform automatic arbitration within a resource domain to enforce one or more policies. One method that may be used to enforce policies 300 is illustrated in FIG. 3. As previously described, monitoring agent 102 monitors 305 one or more application profiles 112-116 for compliance with the policy data 108. If monitoring agent detects a policy violation 310 or anticipated policy violation, one or more actions associated with the policy may be taken to automatically enforce the policy 315. These actions may include expanding a domain by adding more resources to the domain or performing arbitration within a domain. In cases where a policy cannot be enforced, policies may be arbitrated using their associated priorities and a message may be provided to a user that a lower priority policy cannot be met.

[0027] One method that may be used to enforce policies can be described with reference to FIGs. 4A-4D and 5. FIGs. 4A-4D illustrate an exemplary partitioning of a server into multiple partitions. Resource 222 may be partitioned into multiple hardware partitions 302-306. A hardware partition (e.g., Hewlett Packard's nPars) may run its own copy of an operating system and may be electrically isolated from other hardware partitions. One or more of the hardware partitions may provide a container 304 for an application.

[0028] Hardware partition 302 may be further partitioned into processor sets 312, 314, one or more of which may be a container for an application 314.

A processor set may be a resource partition implemented within a single copy of the operating system that contains one or more CPUs. Additionally, hardware partition 306 may be partitioned into multiple software-based partitions 322, 324. A software-based-based partition (e.g., Hewlett Packard's vPars) may be a partition that is implemented in software and has its own copy of the operating system but is not electrically isolated from other software-based partitions. One of the software-based partitions may be associated with a container for an application 324. Another software-based partition 322 may be further divided into sub-CPU resource partitions 332-336 to apportion fractions of CPU resources. One or more of the sub-CPU resource partitions may each be associated with containers 332, 334 to execute an application.

[0029] To enforce a policy 315A associated with an application or container 332, a container 332 may be expanded 505 by reallocating how resources are mapped to the container. Resources may be reallocated by resizing one or more of the partitions. By way of example, container 332 may be expanded to include a larger percentage of the CPU resources. Software-based partition 322 and/or hardware partition 306 may also be resized to enforce one or more policies associated with container 332 or the application running in container 332. Similarly, partitions may be resized at various levels of the server to enforce or meet policies for containers 304, 314, 324, 332, 334, or applications running in the containers.

[0030] An alternate method that may be used to enforce policies 315 can be described with reference to FIGs. 6 and 7. FIG. 7 illustrates a domain that consists of a cluster. The cluster includes two containers 710, 720, each of which is associated with an application. Container 710 includes nodes 712, 714, 716. Container 720 includes nodes 722, 724. By way of example, container 710 may host Web servers and container 720 may be hosting batch workloads. Container 720 may only be currently using node 724 to execute the batch workloads.

[0031] To enforce a policy 315B associated with the Web server application, the monitoring agent 102 may transfer node 722 from container 720 to container 710. Any applications running on node 722 may be shut down so that the Web server application can be instantiated on node 722. The instructions on how to instantiate the Web server application may be located in an application profile 112 associated with the Web server application.

[0032] It should be appreciated that the methods described above may be performed by hardware components or may be embodied in sequences of machine-executable instructions, which may be used to cause a machine, such as a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the actions set forth in FIGS. 3, 5 and 6. Alternatively, the methods may be performed by a combination of hardware and software or the instructions could be partially or wholly integrated into the controller 100 shown in FIG 1.